Attorney Docket No.: 18926-003220US
Client Reference No.: D2317

# PATENT APPLICATION

# OBJECT SECURITY IMPLEMENTATION

Inventor(s):

Robert S. Eisenbart, a citizen of United States, residing at,
833 East J Street
Chula Vista, CA 91910

Annie O. Chen, a citizen of United States, residing at,
12927 Long Boat Way
Del Mar, CA 92014

Patrick J. Murphy, a citizen of United States, residing at,
8631 Al Court
San Diego, CA 92123

Assignee:

GENERAL INSTRUMENT CORPORATION
101 Tournament Drive
Horsham, PA 19044

Entity: Other than a small entity

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 303-571-4000

# OBJECT SECURITY IMPLEMENTATION

This application claims the benefit of U.S. Provisional Application No. 60/165,095 filed on November 12, 1999 and U.S. Provisional Application No. 60/173,963 filed on December 30, 1999.

## BACKGROUND OF THE INVENTION

This invention relates in general to conditional access systems and, more specifically, to authenticating information sent to a set top box.

Cable television (TV) providers distribute content to users. Conditional access (CA) systems distribute video programming from a headend of the cable TV provider to a set top box associated with a user. The headend includes hardware that receives video and distributes it to the set top boxes within the CA system. Select set top boxes are allowed to decode certain video programs according to entitlement information sent by the cable TV provider to the set top box.

Video programs are broadcast to all set top boxes, but only a subset of those boxes are given access to specific video programs. For example, only those that have ordered a pay per view boxing match are allowed to view it even though every set top box may receive the match. Once a user orders the pay per view program, an entitlement message is broadcast in encrypted form to all set top boxes. Only the particular set top box the entitlement message is intended for can decrypt it. Inside the decrypted entitlement message is a key that will decrypt the pay per view program. With that key, the set top box decrypts the pay per view program as it is received in real-time. Some systems sign entitlement messages.

As described above, conventional CA systems only check entitlement of content upon receipt. More sophisticated techniques are always desired to further ensure against content being received from unintended sources.

## SUMMARY OF THE INVENTION

According to the invention, disclosed are an apparatus and methods for authenticating information sent to a set top box. In one embodiment, a method for distributing information that includes a signature is disclosed. In one step the signature is generated over first information and second information. The first information is sent over a network in another step. The second information is sent over the network separately from the step of sending the first information. The signature is sent over the network separately from at least one of the first information and the second information.

In another embodiment, a method for detecting modification of information is disclosed. First and second information are separately received from a network. A signature is received separately from the network from at least one of the first and second information. The signature is authenticated over the first and second information.

In yet another embodiment, a conditional access system for detecting modification of information is disclosed. Included in the conditional access system are an information object and authorization information. A signature is generated over the information object and the authorization information.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing one embodiment of a conditional access system;

Fig. 2 is a block diagram illustrating another embodiment of a conditional access system;

Fig. 3 is a block diagram depicting an embodiment of an authorization message;

Fig. 4 is a block diagram showing an embodiment of a software message;

Fig. 5 is a block diagram illustrating an embodiment of a signatory group that includes portions of the authorization message and the software message;

Fig. 6 is a flow diagram depicting an embodiment of a process for sending an authorization message and a software message to a set top box; and

Fig. 7 is a flow diagram showing an embodiment of a method for processing an authorization message and a software message.

## DESCRIPTION OF THE SPECIFIC EMBODIMENTS

The present invention authenticates information sent to a television set top box. In one embodiment, a software object and a separate data structure containing authorization information are digitally signed with a single signature covering both. The signature is attached to the authorization data structure. The data structure and the software object are sent to the set top box separately. When the set top box receives both, the signature is verified. This process allows authentication of both the data structure and software object in a more robust manner.

In the Figures, similar components and/or features have the same reference label. Further, various components of the same type are distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the several similar components having the second label.

Referring first to Fig. 1, a block diagram of one embodiment of a conditional access (CA) system 100 is shown. The CA system 100 selectively provides content to a number of users based upon certain conditions being satisfied. Included in the system are a headend 104, number of set top boxes 108, local programming receiver 112, and satellite dish 116.

The headend 104 receives content and distributes that content to users. Content can include video, audio, interactive video, software, firmware, and/or data. Distributing firmware content allows updating the embedded programs within the set top box 108. This content is received from a variety of sources which include the satellite dish 116, local programming receiver 112, microwave receivers, packet switched networks, etc. Each set top box 108 has a unique address that allows sending entitlement information to an individual set top box 108. In this way, one set top box 108-1 can entitle a particular content while another 108-2 cannot. Equipment within the headend 104 regulates which set top boxes 108 are entitled to which content.

The content is generally distributed in digital form through an analog channel that contains multiple content streams. All the content streams are statistically multiplexed together into a digital stream modulated upon the analog carrier channel. The separate content streams are separated by packet identification (PID) information such that the individual content streams can be removed according to their unique PID information. There are around one hundred and twenty analog channels in this embodiment of the system 100. Other embodiments could distribute the content with satellite dishes, microwave antennas, RF transmitters, packet switched networks, cellular data modems, carrier current, or phone lines.

Referring next to Fig. 2, a block diagram of another embodiment of a CA system 200 is illustrated. This embodiment 200 shows the software objects 204, 208, 212 that receive the content and distribute it to the set top boxes 108. These software objects 204, 208, 212 could reside in the headed 104. Included in the CA system 200 are a permissions, resource, object signatory (PROS) software 204; a message spooler 208; an object spooler 212; a distribution network 216; and a number of set top boxes 108.

The PROS software 204 receives information from various sources and produces authorization and software messages with that information. Things such as software access requirements, resource access requirements, configuration data, unsigned software objects, object information, etc. are received and processed by the PROS software 204. The resource access requirements indicate which set top box hardware resources can be accessed by the software object. Included in the configuration data is the domain identifier of the cable provider and the signing key(s). The object information contains the identification and version information of the software object. All this information is processed by the PROS software to produce the authorization and software messages. At this point, the messages are complete except for checksums.

The message and object spoolers 208, 212 respectively receive the authorization and software messages and stores them. After adding a checksum, the messages are sent to the set top boxes 108. Even though these messages are broadcast to a number of set top boxes 108, addressing individual domain identifiers and set top box identifiers allows selecting a subset of the set top boxes 108 to receive the messages. The spoolers 208, 212 queue the messages, segment and format them for transport and couple them to the network 216. As the messages are sent out to the network 216, the spoolers 208, 212 calculate a checksum for each message and append that checksum to the end of the message.

The network 216 transports the messages from the message and object spoolers 208, 212 to the set top boxes 108. The network 216 could include a control data channel, MPEG data stream and/or packet switched network. The control data channel is an out of band data channel that contains configuration and control information. Messages sent in a MPEG data stream are statistically multiplexed on an analog carrier channel where the messages are distinguished by unique PIDs. The packet switched network could be the Internet, a network and/or a telephone line connection.

With reference to Figs. 3-5, an authorization message 300, a software message 400 and a signatory group 500 are respectively shown in block diagram form. Included in the authorization message 300 of Fig. 3 are an authorization header 304, an authorization data structure 308, a signature 312, and a first checksum 316. The authorization message 300 has information used to both authenticate and authorize the software message 400. Forming the software message of Fig. 4 are an object header 404, a software object 408 and a second checksum 412. The software message 400 serves as the transport for the software object 408. The signatory group 500 includes components of the authorization message 300 and software message 400 arranged end-to-end. The signature 312 is calculated over the whole signatory group 500. More specifically, the signatory group 500 of Fig. 5 includes the authorization header 304, authorization data structure 308, object header 404, and software object 408.

The authorization header 304 indicates the configuration of the authorization message 300. Included in the header 304 are a subtype identifier and message version. The subtype identifier distinguishes the various types of authorization messages 300 from one another. In this embodiment, there are authorization message subtypes corresponding to software objects and resources. Software object subtypes have a corresponding software message 400, but resource subtypes do not. Accordingly, the subtype identifier is used to determine if there is a software message 400 associated with an authorization message 300. There may be several types of software object subtypes and resource subtypes for a given system and the message version allows distinguishing the various types.

The authorization data structure 308 provides authorization information to the set top box 108. In the case of an authorization message subtype corresponding to a software object, the authorization data structure 308 contains an object identifier, a software version, cost information, entitlement information, lifetime information, and one or more program tiers. The object identifier is unique for each software object 408 and allows attributing an authorization message 300 to its corresponding software message 400. Version information is included in the data structure 308 to indicate the version of the software object 408.

Portions of the authorization data structure 308 are used to determine availability of the software object 408 to the set top box 108. The cost information indicates to the set top box 108, and sometimes the user, the price associated with the software object 408. Entitlement information is used to determine if the particular set top box 108 is authorized to accept the software object 408. The entitlement information may include a key if the software object 408 is encrypted with a symmetric key. If the set top box 108 is not authorized for the software object, there is no need to process the corresponding software object 408 when it is received. Lifetime information allows expiring of the authorization of the software object 408 to prevent use after a certain date or time. Programming tiers are used to restrict authorization of software objects 408 to predefined tiers such that a set top box 108 can only access software objects 408 within a predetermined tier.

The signature 312 is used to verify that portions of both the authorization message 300 and corresponding software message 400 are authentic. A hash function such as SHA-1 or MD5 is run over the whole signatory group, whereafter the result is run through a signing algorithm such as RSA, ECC and DSA to produce the signature. Alternatively, a simple CRC algorithm could be used for the hash function, whereafter the result could be sent through an encryption algorithm such as or triple-DES and DES to produce the signature. When compiling the authorization message 300, the PROS software 204 calculates the signature 312 over the whole signatory group 500 before inserting the signature 312 into the authorization message 300. The set top box 108 calculates the signature of the signatory group 500 upon receipt of both the authorization and software messages 300, 400. Once the signature is calculated, it is checked against the received signature to authenticate portions of both the authorization and software messages 300, 400. If the signatures do not match, the set top box 108 discards the software message 400 because it presumably came from an improper source.

The first and second checksums 316, 412 are calculated with either linear or non-linear algorithms. These checksums 316, 412 verify the integrity of the data as it is transported to the set top box 108 over the network 216. For example, the checksum could be a cyclic redundancy check (CRC) which performs a binary addition without carry for each byte in the message. The message spooler 208 calculates the checksum 316 as the message 300 is being sent and appends the checksum 316 onto the end of the message 300. Conversely, the set top box 108 calculates the checksum as the message 300 is received and checks the calculated checksum against the checksum 316 in the received message 300. If the calculated and received checksums do not match, an error in transmission has occurred. Messages 300, 400 with errors are discarded whereafter the headend 104 may send replacement messages 300, 400.

The object header 404 includes attributes for the software message 400. Included in the object header 404 are a header length, a software object length, the object identifier, the software version, and a domain identifier. The header length and software object length respectively indicate the lengths of the object header 404 and the software object 408. As described above, the object identifier provides a unique code which allows attributing the authorization message 300 to the software message 400. The software version indicates the version of the software object. Different cable providers are assigned domain identifiers such that all of the set top boxes 108, which might receive a software object 408, can screen for software objects 408 associated with their domain.

The software object 408 includes content the system 200 is designed to deliver to set top boxes 108. Several types of information can be embedded in a software object, such as executable programs, firmware upgrades, run-time programs (e.g., Java® or ActiveX®), programming schedules, billing information, video, audio, or data. The software object can be used immediately after authentication and authorization or at a later time. Additionally, authorization can be programmed to expire after a certain amount of time.

Referring specifically to Fig. 5, the signatory group 500 is shown. This group 500 is comprised of parts of both the authorization message 300 and the software message 400. All the data used to calculate the signature 312 is included in the signatory group 500. Because the signature requires components from both the authorization message 300 and the software message 400, a failed signature check indicates one of the authorization message 300 and the software message 400 cannot be verified as originating from a trusted source. The trusted source being the PROS software 204 that generated the signature 312.

With reference to Fig. 6, a flow diagram of a process for broadcasting a software object 408 is shown. This process uses the PROS Software 204 to compile the information and create a signature whereafter the authorization message 300 and corresponding software message 400 are sent to their respective spoolers 208, 212. Preferably, the messages 300, 400 are sent through different transmission channels at different times to make authentication more robust.

The process begins in step 604 where the software object 408 and other information are loaded into the PROS software 204. This information includes such things as software access requirements, resource access requirements, configuration data, unsigned software objects, object information, etc. In step 608, the PROS software 204 determines the components of the signatory group 500. Once the signatory group 500 is determined, the signature 312 over that group 500 is calculated in step 612. More than one signature could be calculated in this step 612 to support different set top boxes 108, as explained further below.

Once the signature 312 is calculated, the authorization and software messages 300, 400 are created in step 616. At this point, the authorization and software messages 300, 400 are complete except for the checksums 316, 412. In step 620, the authorization and software messages 300, 400 are respectively sent to the message and object spoolers 208, 212. Once the spoolers 208, 212 receive the authorization and software messages 300, 400, the checksums 316, 412 are calculated to complete all fields in the messages 300, 400.

After the authorization and software messages 300, 400 are complete, they are separately sent to the set top boxes. In step 628, the authorization message 300 is broadcast to the set top boxes 108 over a network 216. The network 216 could include a control data channel, MPEG data stream and/or packet switched network. After the authorization message 300 is sent, the software message 400 is broadcast in step 632. A predetermined time delay may be used between steps 628 and 632 to separate the messages 300, 400. In this embodiment, the messages 300, 400 are sent through different transmission pathways. The authorization message 300 is sent over an out-of-band control data channel and the software message 400 is sent through a statistically multiplexed MPEG data stream.

Referring next to Fig. 7, a flow diagram is shown that verifies, authenticates and authorizes a software object 408. This process is performed on each set top box 108 that receives the authorization and software messages 300, 400 even if the software object 408 is not authorized and cannot be executed.

The process begins in step 704 where the authorization message 300 is received from the network 216. After receipt of the message 300, the first checksum 316 is verified to trap broadcast errors. A threshold determination is made to decide if the data structure 308 is authorized in step 708. Things such as cost information, entitlement information, lifetime information, and a program tier is used in this determination. If it is determined that the software object 408 is not authorized, it is ignored when it arrives in step 712.

Alternatively, processing continues to step 716 if the software object 408 is determined authorized in step 708. In step 716, the software message 400 is received and the second checksum 412 is verified to trap any transmission errors. The object identifiers in the authorization and software messages 300, 400 are compared to match together the messages 300, 400 in step 720. If there is no match, the software message 400 is discarded and processing continues back to step 716 to wait for the correct software message 400.

If the software message 400 has an object identifier which matches the object identifier of the authorization message 300, processing continues to step 724 where the information is arranged into a signatory group 500. Most of the information in the authorization and software messages 300, 400 is used to form the signatory group 500. In step 728, a signature is calculated over the signatory group 500. Once the signature is calculated, it is compared with the received signature 312 in step 732.

If the calculated and received signatures match, as determined in step 736, the software object 408 is authenticated as originating from an approved source. Authenticated software objects 408 are retained and used by the set top box in step 740. If the software object fails authentication in step 736, the software message 400 is discarded and an error is reported back to the headend 104. By using this process, software objects are verified, authorized and authenticated before use.

In light of the above description, a number of advantages of the present invention are readily apparent. Authentication is performed on both the authorization message and the associated content. The authentication is more robust since the authorization message and content are sent to the set top boxes separately such that a pirate would have to know how to reassociate the components in any effort to thwart authentication.

A number of variations and modifications of the invention can also be used. For example, different signature methods could be used which include symmetric and asymmetric using different key sizes. Some embodiments could encrypt the authorization and software messages to further enhance authentication and use entitlement messages to provide the keys.

Some of the above embodiments use a single signature in the authorization message. However, different set top boxes use different signature algorithms and/or keys. To make an authorization message that is understood by all set top boxes, the authorization message can include a number of signatures and each set top box chooses the compatible signature when authenticating the messages.

Sub C1 >

~~Although the above embodiments attach the signature to the authorization~~ message, other possibilities are considered within the scope of this invention. For example, the software message could include the signature. Alternatively, the signature could be send separately from both the authorization message and software message. The common thread is that some of the information for which the signature is generated over is sent without the signature. In this way, the signature is disassociated from some information such that a hacker would have to know how to reassociate the information to ~~have the signature match.~~

Although the invention is described with reference to specific embodiments thereof, the embodiments are merely illustrative, and not limiting, of the invention, the scope of which is to be determined solely by the appended claims.